JIKO (Jurnal Informatika dan Komputer) Accredited KEMENDIKBUD RISTEK, No.105/E/KPT/2022
Vol. 7, No. 2, August 2024, pp. 131-137 p-ISSN: 2614-8897
DOI: 10.33387/jiko.v7i2.8539 e-ISSN: 2656-1948

DESIGN OF MICROSLEEP DETECTION SYSTEM IN 32 BIT MICROCONTROLLER-BASED MOTORISTS WITH RANDOM FOREST METHOD

Syiva Awaliyah Maqdis [1], Anugrah Adiwilaga [2], Munawir Munawir[3]

[1,2,3] Teknik Komputer, Universitas Pendidikan Indonesia, Indonesia
[*] Email: [1] maqdissyiva@upi.edu, [2] anugrah.adiwilaga@upi.edu, [3] munawir@upi.edu

(Received: 29 July 2024, Revised: 15 August 2024, Accepted: 22 August 2024)

Abstract

The number of motorcycle accidents has increased rapidly every year. Many occur due to drowsiness or fatigue because motorists force themselves to keep driving. The state of fatigue while driving is also known as microsleep. To overcome this problem, we propose a design of a prototype system that can be installed on the helmet of a motorized user so that the driver is more alert when driving a vehicle. This system utilizes machine learning technology with the Random Forest algorithm with two prediction results: prediction 1, which means the motorcyclist is tired, or prediction 0, which means the motorcyclist is in a normal state, embedded in the ESP32 microcontroller, and a tilt sensor that can detect signs of drowsiness in motorists. This system design will use the MPU6050 sensor to measure changes in the angle of the motorcyclist's head. The microcontroller will process the data obtained to identify head changes that indicate the possibility of drowsiness. If it occurs, the buzzer will beep as a warning to warn the driver to take a short break. The test results in drowsiness conditions with an angle of 10°–30° resulted in 100% accuracy, and normal conditions only at an angle of 0°–6° resulted in 100% accuracy. The result of the developed system is expected to reduce the number of accidents caused by drowsiness.

Keywords: *Microsleep, Prototype, Microcontroller, MPU6050, Random Forest*

**Corresponding Author: Syiva Awaliyah Maqdis*

1. INTRODUCTION

Traffic accidents continue to be a serious problem that threatens lives. According to data from the Badan Pusat Statistik (BPS), the number of accidents in Indonesia from 2019-2021 is high. Safety in mobilization is a crucial aspect that receives significant attention. One of the factors that cause accidents when driving is drowsiness, which society often ignores [1]. Motorists who force themselves to drive when they are excessively drowsy can reduce the driver's alertness and can cause traffic accidents that endanger other people and the driver himself.

The occurrence of fatigue when driving to the point of feeling drowsiness is also known as microsleep. Microsleep is an event where a person experiences a loss of consciousness for just a fraction of a second, and it can happen at any time. Microsleep events do not last long, only seconds to 2 minutes. If a person experiences closed eyes for 0.5 seconds or more periodically, every 10 seconds or so, this indicates that the person is experiencing microsleep. Microsleep makes a person fail to focus, ignoring the surroundings alike as sensory input that later enters a person's subconscious [2].

When humans reach adulthood or are calculated starting from the age of 17 years and above, they need sleep time for approximately 8 hours every night. This microsleep factor is caused by a lack of sleep, making the body tired [3]. Since drowsiness is a part of microsleep, we can detect the microsleep condition from the drowsiness. Drowsiness is a condition of a person between conscious and unconscious that causes a decrease in function in all five senses so that the level of a person's head tilts when drowsiness happens, is 15-45 degrees within 3 seconds [4], [5]. Drowsiness can be categorized as the condition of the eyelids starting to be heavy, the view beginning to blur, and the head starting to be unbalanced [6]. Several methods have been found to carry out early detection when someone is drowsiness, one of which is the head position change method [7].

People should be more aware of the incidence of microsleep. Uniquely, microsleep does not only occur due to lack of sleep; there are several cases in which microsleep occurs in individuals who perform monotonous activities. What needs to be understood about microsleep is its impact or risk. An example of the effects that must be considered is fatigue, a decrease in a person's productivity performance when doing activities. Loss of alertness and awareness of the surrounding environment so as not to be able to detect when there is danger can also cause a person to have a social relationship disorder that makes a person not interested in social activities [8], [9], [10].

Technology has played a crucial role in developing practical solutions to keep drivers safe from the dangers and impacts of microsleep. The primary goal is to improve traffic safety for motorcyclists, thereby reducing the risk of accidents caused by microsleep while driving. Another objective of this research is to create a drowsiness warning system integrated into the motorcyclist's helmet. This system detects signs of drowsiness in the motorcyclist and provides warning signals to prevent them from entering a state of microsleep.

Several studies have explored the dangers of microsleep, leading to an innovation in creating a system to detect drowsiness in drivers. The research aims to detect drivers in a state of microsleep through eye blinks and eye aspect ratio combined with machine learning algorithms [11]. Another study used MPU6050 sensors to detect the tilt of an accelerometer and a gyroscope to enter the reference points of roll tilt, yaw, and pitch stored on a motorcycle seat or trunk [12]. In addition, some studies use the classification of Random Forest to detect human movement during activities such as climbing stairs, walking, talking, standing, and working [13]. Therefore, an innovation was formed to prevent accidents caused by microsleep. This innovation utilizes the Lolin32 Lite, a microcontroller or component controller that connects to other components. The system is enhanced by integrating the MPU6050 sensor and a Buzzer. Lolin32 Lite has two cores that allow the microcontroller to work faster [14]. The system combines machine learning, microcontroller, and sensor technology to detect early signs of fatigue or drowsiness in drivers and provide a quick warning to reduce the risk of accidents. Machine learning is also a part of Artificial Intelligence, usually used to answer a possibility that can only be imagined [15], Consequently, the device will be implemented with machine learning using the Random Forest algorithm, and accurate results will be expected to distinguish drowsiness and non-drowsiness positions. Random Forest is an ensemble algorithm based on bootstrap aggregating, known for its superior performance compared to other Decision-Tree algorithms [16], [17].

This MPU6050 sensor is one type of sensor with the MEMS (Micro-Electro-Mechanical System) type, where this sensor is widely applied to detect attitudes in robots to energy harvesting [18]. MPU6050 sensor is widely used due to its ability to detect changes in movement and orientation of the helmet, which can indicate drowsiness in the motorcyclist. Information from the gyro sensor MPU6050 was obtained by reading the angular tilt based on the input data, which converts the acceleration and rotational movement data into digital data [19]. The gyro sensor then integrates the angle value into machine learning, sending a signal to the Lolin32 Lite to obtain the appropriate sensor output value and distinguish between head nods and head shakes. To make the design function optimally, we have to identify and consider the limit condition of the road. However, this system is specifically designed for paved roads with flat and smooth surfaces. In addition, when the system detects a drowsy head position, the buzzer sounds within 2-3 seconds to provide timely warnings. Initial testing focuses on smooth road conditions to ensure the tool's effectiveness. By addressing this limitation, the design is expected to perform well in various real-world road conditions and contribute to reducing motor vehicle accidents caused by drowsiness.

2. RESEARCH METHOD

This research uses the waterfall method, namely by taking an approach in software development that describes the process in a linear sequential manner. The waterfall method will carry out system design sequentially, starting with identification and analysis according to needs to testing and maintenance.

2.1 Identification and Analysis

In this section, needs identification is carried out by creating a prototype system design, which will be tested by attaching the prototype to a helmet. After testing, the existing data will be integrated using the Random Forest machine learning algorithm. Once the data has been read, it will sent to the Lolin32 Lite for processing. The buzzer will signal the motorcyclist if drowsiness is detected. Otherwise, it will continue to monitor the head continually.

We use the Random Forest as a machine-learning algorithm to combine the results of many decision trees to achieve one result. Each tree is built using a random subset of the dataset already obtained by retrieving data to measure a random subset of features from each partition. After that, the Random Forest will make predictions to combine the results of all the trees [20] [21], [22]. The advantages of Random Forests are that they can incorporate many models, cope with complex data, are more robust, and reduce the risk of overfitting. Because they use ensemble averaging, Random Forests are more resistant to overfitting. They are also more stable because they take the average of many models from introducing variability between individual trees, improving overall prediction performance. The Random Forest work stages used in this research include sampling, building decision trees, prediction, and final result output.

A. Sampling

1. Start by taking several random samples to form the training set.

2. The samples that have been taken will be used to build a Decision Tree.

B. Building a Decision Tree

1. Split the dataset into a sleepy and normal subset.
2. Each internal node in the Decision Tree represents a feature or attribute, the branches represent decision rules, and the leaves represent results or labels.

C. Prediction by Voting

1. After the Decision Tree is built, Random Forest will make predictions on the test data.
2. Each Decision Tree will provide its prediction, where prediction 1 is identified as sleepy and prediction 0 is identified as normal.
3. A voting process is carried out for each prediction result that comes out of all Decision Trees.

D. Final Result

1. Random Forest will choose the most voted prediction result as the final prediction.

2.2 Design

A. Block Diagram

In this research, we developed a device that can warn motorcyclists if their microsleep condition is detected. The block diagram, shown in Figure 1, explains the device's components. We used ESP32 as our main processor in the form of a Lolin32 Lite board. ESP32 is connected to an MPU6050 and other components, such as a button, buzzer, LED, and battery. The device that we developed is designed to be attached to a motorcyclist's helmet.

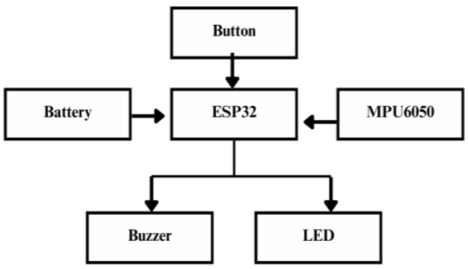

Figure 1. Block Diagram

B. Flowchart

Figure 2 will explain the flow. When the device has started, the MPU6050 will read the helmet's attitude by giving the accelerometer and gyro values. Then, the ESP32 will receive data from MPU6050, and then the ESP32 will process with a machine learning algorithm. The algorithm will predict and output a result of 0 (normal condition status) or 1 (drowsy condition status). The ESP32 commands the buzzer according to the prediction of head tilt conditions. If the prediction result is 1, the buzzer will turn on. Otherwise, with 0 output, the system will continue re-reading the accelerometer and gyro values. So, the system is expected to warn if drowsiness is detected.

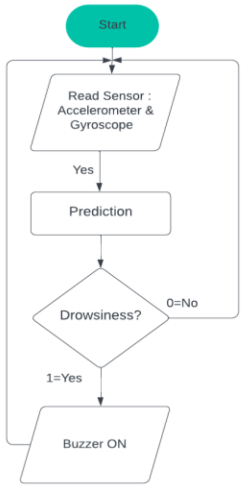

Figure 2. Flowchart of drowsiness detection device microcontroller program

C. Wiring Diagram

Figure 3, the wiring diagram, illustrates the components connected in one circuit.

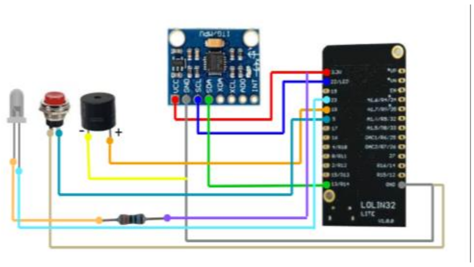

Figure 3. Wiring Diagram

D. Component Prototype

Figure 3 shows the wiring between the components connected between Lolin32 Lite, MPU6050, buzzer, LED, and button.

a. Pin GND Lolin32 Lite to GND MPU6050
b. Pin 3V Lolin32 Lite to VCC MPU6050
c. Pin 22 Lolin32 Lite to SCL MPU6050
d. Pin 13 Lolin32 Lite to SDA MPU6050
e. Pin 18 Lolin32 Lite to + Buzzer
f. Pin GND MPU6050 to - Buzzer
g. Pin 5 Lolin32 Lite to Push Button
h. Pin GND Lolin32 to Push Button
i. Pin 23 Lolin32 Lite to LED Cathode
j. Pin 3V Lolin32 Lite to Resistor
k. Anode LED to Resistor

E. Illustration of Head Attitude

Figure 4 explains the attitude or tilting variation of motorcyclists. In a drowsiness state, the initial position of the head will start tilting forward compared to the upright position [23].

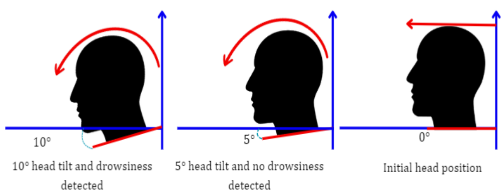

Figure 4. Illustration of head tilt

F. Illustration of the attached device

The direction of the axis on the device after being attached to the helmet is in the roll direction (the x-axis points to the front and back), because the initial position of drowsiness is leaning towards the front. As shown in Figure 5, the device is attached to the left side of the helmet, and the x-axis will detect when the motorcyclist is in a state of drowsiness by giving a warning with a buzzer sound.

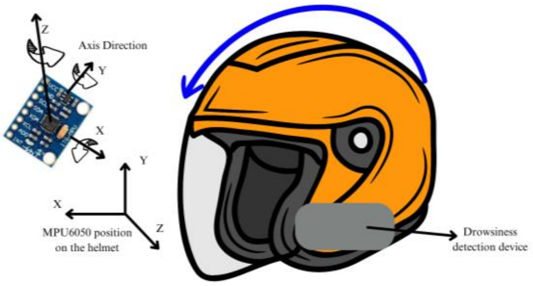

Figure 5. Illustration of Axis MPU6050 on helmet

G. Research structure

Figure 7 explains the flow of this research which is divided into two parts, namely journal writing, and prototype. In writing, carry out a literature review to find journal references to be analyzed and tested, then compare the results of the literature review with the results of the trials that have been carried out to reach conclusions.

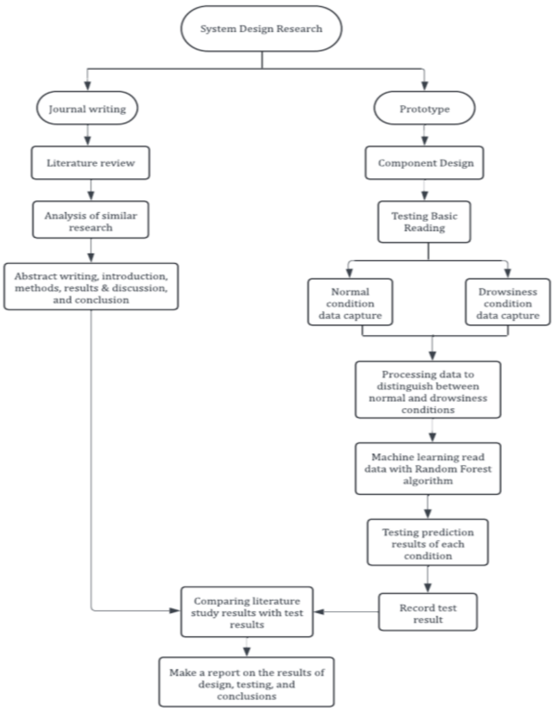

Figure 7. Research Structure

2.3 Implementation

The prototype device is mounted on the left helmet with a switch and push buttons, as shown in Figure 8. The switch button turns the detection device ON and OFF when it is in use and not in use. We provided a feature to pause the device using a push button. When the detection device is running, the LED indicator lights up and turns off when the device is paused.

2.4 Testing

Testing is carried out with four stages, namely, taking a drowsy condition dataset, taking a normal condition dataset, testing in drowsy conditions with a slope of 10°, 15°, 20°, 25°, and 30° with 10 tests each, and testing in normal conditions with a slope of 0°, 2°, 4°, 6°, and 8° with 10 tests each.

The datasets used for training in Random Forest are taken as many as two kinds: drowsiness datasets and normal datasets. So, there are 30 sets of drowsiness data and 30 sets of normal data. Each 1 dataset has 6 values (accelerometer X, accelerometer Y, accelerometer Z, gyro X, gyro Y, and gyro Z) for a total of 180 datasets.

Next, machine learning will study the results of the datasets taken. Then, training is carried out using the Random Forest algorithm, which transfers the model to the ESP32 for the interference process (prediction).

2.5 Maintenance

For future maintenance, it is necessary to periodically check the battery and be careful when it rains heavily. In the detection device, a push button is also prepared for the driver to use if the driver is doing activities other than driving. This is done so that the battery can last a long time and is not continuously charged. The battery can be charged when it has been used for 2-3 days.

3. RESULT AND DISCUSSION

In this section, we will present our work's results, including the design implementation and also several tests carried out to evaluate the performance. The device's appearance can be seen in Figure 7, Figure 8, and Figure 9. Figure 7 shows installed components inside the device through the casing from the top view. All of the installed components are connected via the PCB-Board. The components were fit in compact form, including a LiPo Battery on the side, Lolin32 Lite, a Buzzer stored at the bottom of Lolin32 Lite, an LED on the side, and a Push Button. We can access the ON-OFF and push buttons from the outside of the casing, which are placed next to each other.

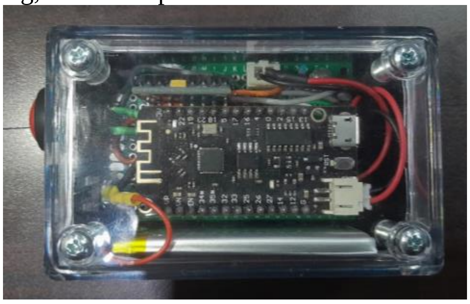

Figure 7. Top View Prototype Component

Figure 8. Side View Prototype Component

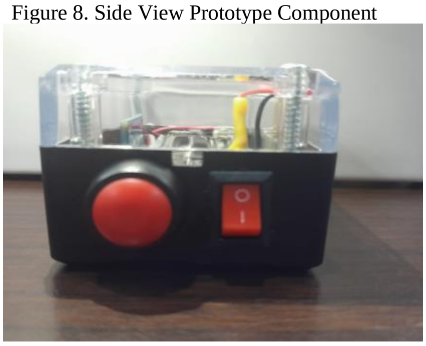

Figure 9. Front View Prototype Component

Once the components are stored in the casing, the device is ready for use by the motorcyclist and stored on the left side of the helmet, as seen in Figure 10.

Figure 10. prototype mounted on the left helmet

It will display the test results between sleepy and normal conditions. The sleepy condition here is identified by the direction of the head leaning forward or the direction of the head tilt, namely downward, as illustrated in Figure 4. If the motorcyclist is sleepy, a prediction will appear with a value of 1; if the motorcyclist is in a normal state, a prediction will appear with a value of 0. Therefore, the Random Forest algorithm was chosen to process the data taken during the trial, namely with the result that the head tilts when sleepy is leaning forward with a slope of 10 °; this was obtained from the trial carried out in a sleepy state. Random Forest can read the sleepy value and make the 10 ° slope the beginning of the sleepy position; if the head is in that slope, the buzzer will sound..

The test results display the results of normal conditions (named prediction 0), the percentage of success from each angle tested, and the percentage of buzzer activation remaining in the OFF state. There are 5 angles tested, namely 0°, 2°, 4°, 6°, and 8° with each doing 10 tests. The test results are listed in Table 1.

The test results show the drowsiness condition (named prediction 1), the percentage of success from each angle tested, and the percentage of buzzer activation remaining in the OFF state.

Table 1 . Table 1. Testing results of 0 device prediction against normal conditions

Slope	Normal Condition Prediction (%)	Number of successful tests	Buzzer OFF (%)	Number of successful tests
0°	100%	10	100%	10
2°	100%	10	100%	10
4°	100%	10	100%	10
6°	100%	10	100%	10
8°	90%	9	90%	9

Table 2 shows the results of the angles tested by conducting five trials at different angles, namely 10°, 15°, 20°, 25°, and 30° with each test conducted ten times.

Table 2. Test results of 1 device's prediction of drowsiness condition

Slope	Drowsiness Condition Prediction (%)	Number of successful tests	Buzzer ON (%)	Number of successful tests
10°	100%	10	100%	10
15°	100%	10	100%	10
20°	100%	10	100%	10
25°	100%	10	100%	10
30°	100%	10	100%	10

Furthermore, testing using the Random Forest algorithm will produce an output of 1.0 which means that the test carried out produces perfect results. As in Figure 11, the cross-validation score is 1.0

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=100)

# Membuat KFold dengan 5 splits
kf = KFold(n_splits=5, shuffle=True, random_state=1)

# Melakukan K-fold cross-validation
scores = cross_val_score(model, X, y, cv=kf)

print(f'Skor untuk setiap fold adalah: {scores}')
print(f'Rata-rata skor cross-validation: {scores.mean()}')

Skor untuk setiap fold adalah: [1. 1. 1. 1. 1.]
Rata-rata skor cross-validation: 1.0
```

Figure 11. Results on Jupyter

In the test, the results read by the program were normal conditions and drowsiness conditions. The value that will be read by the program when it is under normal conditions is Prediction 0, and the drowsiness condition is Prediction 1. Figure 12 is an example of the program reading that the motorcyclist is in normal condition.

```
from sklearn.metrics import accuracy_score
skor_akurasi = accuracy_score(y_test, y_pred)

print(f'Skor Akurasi: {skor_akurasi}')

Skor Akurasi: 1.0

# Prediksi label untuk data baru, misal prediksi data input baris 45
X_new = df.iloc[45,:-1].to_numpy()
y_pred_new = knn.predict([X_new])

print(f'Hasil Prediksi: {y_pred_new}')

Hasil Prediksi: [0]
```

Figure 12. Figure 12. Prediction Results on Jupiter

Figure 13 shows that testing with 18 test data resulted in a precision value of 1.00, recall of 1.00, F1-Score of 1.00, and accuracy of 1.00, which means success without error. In machine learning data analysis, 1.00 refers to the highest value of an evaluation matrix.

```
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred))

Confusion Matrix:
[[ 8  0]
 [ 0 10]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         8
           1       1.00      1.00      1.00        10

    accuracy                           1.00        18
   macro avg       1.00      1.00      1.00        18
weighted avg       1.00      1.00      1.00        18
```

Figure 13. Precision, Recall, F1-Score, Accuracy

The test shows prediction 1 (drowsiness condition) and prediction 0 (normal condition) by testing from various degree angles, each carried out 10 times. The test results in drowsy conditions with an angle of 10°-30° resulted in 100% accuracy, and in normal conditions, only the angle of 0°-6° resulted in 100% accuracy.

4. CONCLUSION

Based on the research results, it can be concluded that the microsleep detection device can detect head tilt when drowsiness occurs. This can be proven when the head tilts at the roll angle reaches 10°, and the buzzer will sound. So, when the motorcyclist is in a state of drowsiness, the device will sound the buzzer attached to the left helmet, so that the motorcyclist can immediately wake up when in a state of microsleep. The MPU6050 can detect when the motorcyclist experiences one of the drowsiness characteristics in the form of a forward head nod with an angle of at least 10° and within 2-3 seconds.

5. REFERENCE

[1] S. A. Khan, H. Mukhatar, and B. A. Pramudita, "Perancangan Sistem Pendeteksi Microsleep Untuk Peringatan Kelelahan Pada Pengemudi Kendaraan," *EProceedings Eng.*, vol. 9, no. 4, p. 1810, Aug. 2022.

[2] A. Hertig-Godeschalk, J. Skorucak, A. Malafeev, P. Achermann, J. Mathis, and D. R. Schreier, "Microsleep episodes in the borderland between wakefulness and sleep," *Sleep*, p. zsz163, Jul. 2019, doi: 10.1093/sleep/zsz163.

[3] Andre Hartoko Aji Putra Perdana, Susijanto Tri Rasmana, and Heri Pratikno, "Implementasi Sistem Deteksi Mata Kantuk Berdasarkan Facial Landmarks Detection Menggunakan Metode Regression Trees," *J. Technol. Inform. JoTI*, vol. 1, no. 1, pp. 1–9, Oct. 2019, doi: 10.37802/joti.v1i1.1.

[4] S. Sugeng and T. N. Nizar, "Deteksi Aktivitas Mata, Mulut Dan Kemiringan Kepala Sebagai Fitur Untuk Deteksi Kantuk Pada Pengendara Mobil," *Komputika J. Sist. Komput.*, vol. 12, no. 1, pp. 83–91, May 2023, doi: 10.34010/komputika.v12i1.9688.

[5] N. W. Yanto, L. Elisana, S. W. Lestari, and W. Handini, "Rancang Bangun Sistem Pendeteksi Kantuk Menggunakan Sensor Imu dan WeMOS," *J. Teknol.*, vol. 9, no. 1, pp. 72–79, Nov. 2021, doi: 10.31479/jtek.v9i1.136.

[6] C. Aj. Saputra, D. Erwanto, and P. N. Rahayu, "Deteksi Kantuk Pengendara Roda Empat Menggunakan Haar Cascade Classifier Dan Convolutional Neural Network," *JEECOM J. Electr. Eng. Comput.*, vol. 3, no. 1, pp. 1–7, Apr. 2021, doi: 10.33650/jeecom.v3i1.1510.

[7] D. N. Aditya, C. Nugraha, and H. Prassetiyo, "Perancangan Alat Deteksi Dini Kondisi Kantuk untuk Mengurangi Risiko Kecelakaan Kerja Berbasis Pengolahan Citra Digital.pdf." e-Proceeding FTI, 2022.

[8] F. Damarjati, "Model Mitigasi Kecelakaan Transportasi Menggunakan Pencegah Micro-Sleep," vol. 2, 2022.

[9] B. I. S. Duna, Y. Iashania, and I. Tambang, "Manajemen Fatigue Untuk Mencegah Microsleep Pada Driver Sarana," vol. 23, 2023.

[10] R. P. Firdaus Ariandika Nadapdap, J. Muhamad Jabbar, and M. Pamungkas, "Faktor Risiko Microsleep Pada Driver Ojek Online di Antapani Tahun 2023," STIKES Dharma Husada, Bandung, 2023.

[11] H. Z. Ilmadina, D. Apriliani, and D. S. Wibowo, "Deteksi Pengendara Mengantuk dengan Kombinasi Haar Cascade Classifier dan Support Vector Machine," *J. Inform. J. Pengemb. IT*, vol. 7, no. 1, pp. 1–7, Jan. 2022, doi: 10.30591/jpit.v7i1.3346.

[12] A. Suprayogi and H. Fitriyah, "Sistem Pendeteksi Kecelakaan Pada Sepeda Motor Berdasarkan Kemiringan Menggunakan Sensor Gyroscope Berbasis Arduino," *J. Pengemb. Teknol. Informasi Dan Ilmu Komput.*, vol. 3, no. 3, pp. 3079–3085, Mar. 2019.

[13] G. G. Putra and D. U. Suwarno, "Pembaca Aktivitas Manusia Dengan Sensor Gyro," vol. 1, no. 1, 2019.

[14] S. Edy and A. Hamzah, "Rancang Bangun Prototype Cruise Control Pada Kendaraan Listrik

Dengan Metode Kendali PID," ISTN, Reseacrh Report, 2023.
[15] M. T. D. Putra, A. Adiwilaga, A. Clarissa, A. A. P. Gustiansyah, A. D. Kurniadi, and Z. Mumtaz, "Alat Bantu Tuna Netra Berbasis Arduino Uno dan Artificial Intelligence dengan metode YOLO v7," *J. Ilm. FIFO*, vol. 15, no. 2, p. 159, May 2024, doi: 10.22441/fifo.2023.v15i2.007.
[16] F. A. Rafrastaraa, R. A. Pramunendar, D. P. Prabowo, E. Kartikadarma, and U. Sudibyo, "Optimasi Algoritma Random Forest menggunakan Principal Component Analysis untuk Deteksi Malware," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 5, no. 3, pp. 217–223, Jul. 2023, doi: 10.47233/jteksis.v5i3.854.
[17] I. L. Mulyahati, "Implementasi Machine Learning Prediksi Harga Sewa Apartemen Menggunakan Algoritma Random Forest Melalui Framework Website Flask Python (Studi Kasus: Apartemen di DKI Jakarta Pada Website mamikos.com)".
[18] M. Munawir, D. R. Agustini, R. Rahmawati, and A. M. N. Anzhar, "Machine Diagnosis based Multi-Agent Technology by Autonomous Sensor with Energy Harvesting," *J. Comput. Eng. Electron. Inf. Technol.*, vol. 1, no. 1, pp. 51–62, Apr. 2022, doi: 10.17509/coelite.v1i1.43818.
[19] F. Mangkusasmito, D. Y. Tadeus, H. Winarno, and E. Winarno, "Peningkatan Akurasi Sensor GY-521 MPU-6050 dengan Metode Koreksi Faktor Drift," *Ultima Comput. J. Sist. Komput.*, vol. 12, no. 2, pp. 91–95, Nov. 2020, doi: 10.31937/sk.v12i2.1791.
[20] A. N. Syafia, M. F. Hidayattullah, and W. Suteddy, "Studi Komparasi Algoritma SVM Dan Random Forest Pada Analisis Sentimen Komentar Youtube BTS," *J. Inform. J. Pengemb. IT*, vol. 8, no. 3, pp. 207–212, Sep. 2023, doi: 10.30591/jpit.v8i3.5064.
[21] Universitas Singaperbangsa Karawang and S. Syihabuddin Azmil Umri, "Analisis dan Komparasi Algoritma Klasifikasi Dalam Indeks Pencemaran Udara di DKI Jakarta," *JIKO J. Inform. Dan Komput.*, vol. 4, no. 2, pp. 98–104, Aug. 2021, doi: 10.33387/jiko.v4i2.2871.
[22] G. A. Sandag, "Prediksi Rating Aplikasi App Store Menggunakan Algoritma Random Forest," *CogITo Smart J.*, vol. 6, no. 2, pp. 167–178, Dec. 2020, doi: 10.31154/cogito.v6i2.270.167-178.
[23] M. Amirullah and H. Kusuma, "Sistem Peringatan Dini Menggunakan Deteksi Kemiringan Kepala pada Pengemudi Kendaraan Bermotor yang Mengantuk," vol. 7, no. 2, 2018.